NCRP41

PCIbus CANbus Interface Board
w/4 Channels and 1MB RAM

USER'S MANUAL

Rev 1.1

© CJB 2012
www.cjb.it ~ info@cjb.it

Index

OVERVIEW 3

Main Features: 3

LAYOUT (1) 4

LAYOUT (2) 5

CANBUS CONNECTORS 6

TERMINATIONS 6

BATTERY OPERATION 7

PROGRAMMING 8

Configuration space 8

DETAIL OF REGISTERS 9

PCI Address Space 0 and 1 9

PCI Address Space 2 10

PCI Address Space 3 10

Interrupts 10

Local Registers 10

INTCSR (4Ch) 11

CNTRL (50h) 12

EEPROM contents 13

Can controller 13

Memory 14

CABLE ADAPTOR TO CHANGE RJ45 TO DSUB9M 15

OVERVIEW

The CRP family boards are all PCI short format boards. They conform to the PCI standard V2.1 with target only capabilities. The boards offer up to four independent and optoisolated CAN channels, eight TTL digital inputs, eight TTL digital outputs, and up to 1Mbyte battery backed RAM.

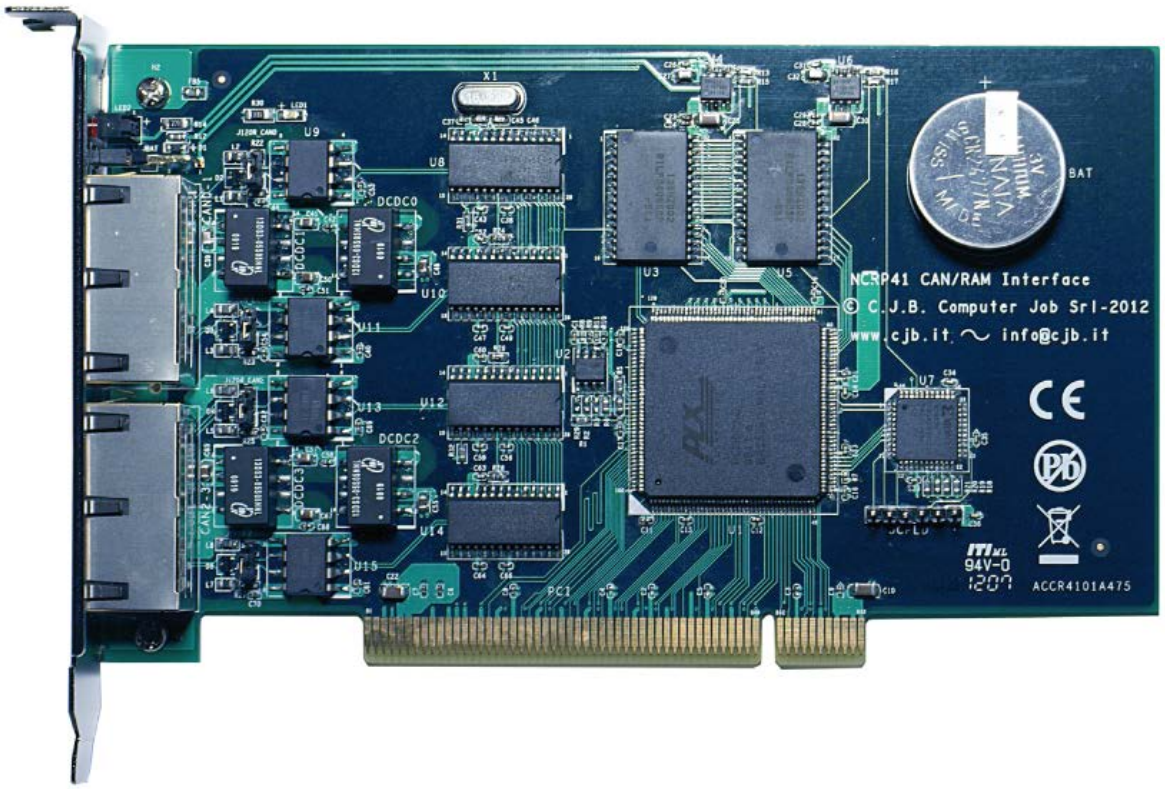
Main Features:

- 4 CANbus ports, isolated (500Vac), with state-of-the-art ISO1050DUBR CAN-drivers (which integrate isolation DCDC)
- selectable termination for each channel
- 1MB Static RAM for unlimited write cycles of system's variables, which is a must for applications in SoftPLC & SoftMotion systems
- 5Vdc power supply
- PCI bus interface 2.1
- 4-layer pcba, high quality manufacturing
- tough 950mAh onboard Lithium battery for long term data preservation in the RAM
- RJ45 standard CANbus ports
- LED for low battery advise
- full software support for: Windows, Linux, QNX

Following pages will lead you inside the board’s architecture and use.

© 2012

LAYOUT (1)



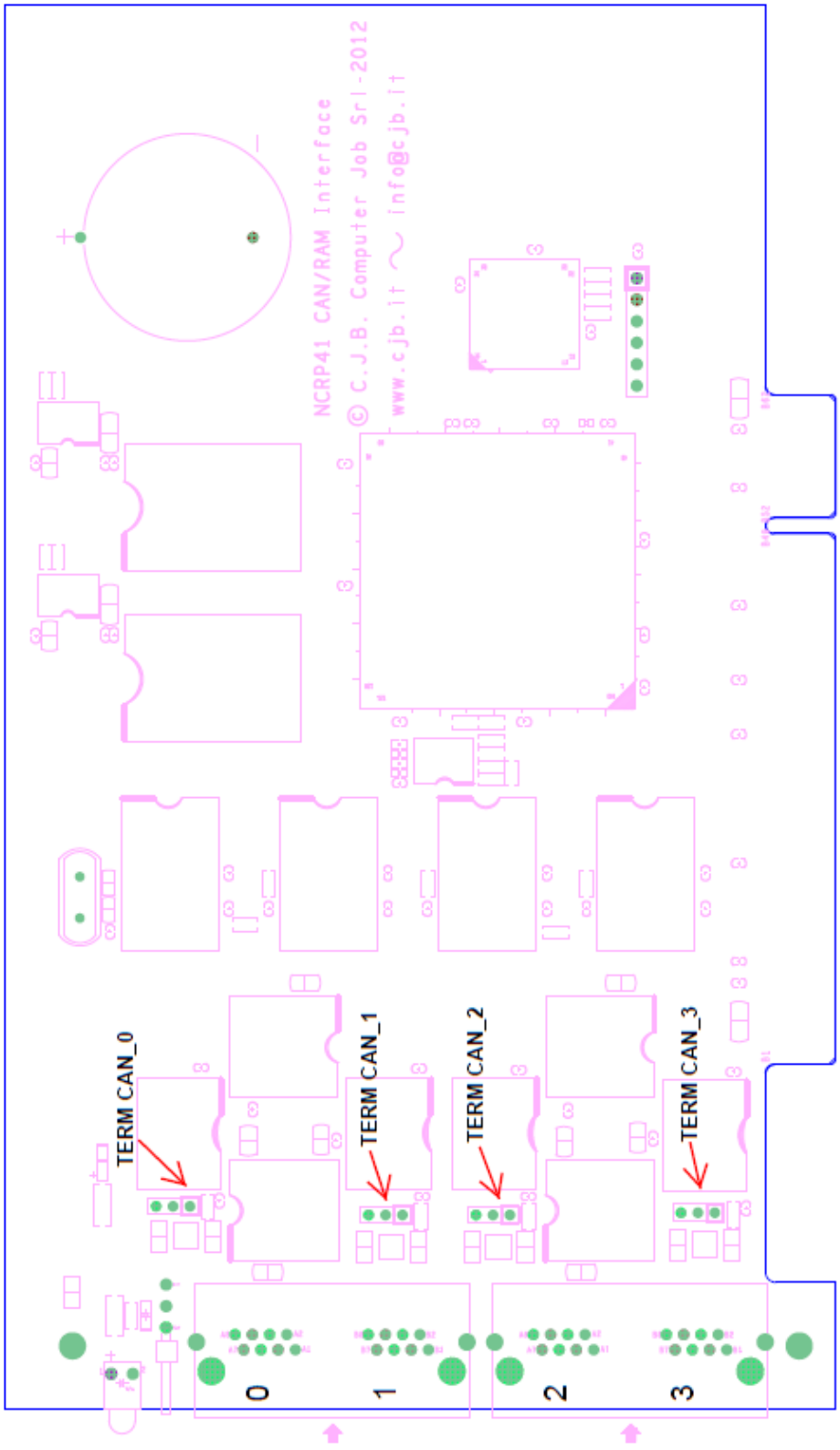
This is a close photograph of the NCRP41 PCIbus add-on card.

Look at the two blocks, each 2xRJ45 (CANbus connectors), the 4 DC/DC for making the isolated power supply to CANbus drivers (isolation: 500Vdc), the 4 SJA1000 CAN controllers, the 2 x 512kB Static RAMs, the big PLX PCIbus interface chipset, the FPGA for board settings & glue logic, the 950mA/h battery.

Upper left is the LED which lights on if battery is low.

Near the LED is the Battery connect/disconnect Jumper.

LAYOUT (2)

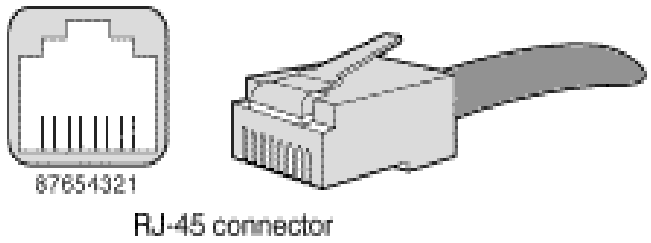


In this layout it is possible to identify the four CAN bus connectors (0, 1,2,3), the four termination jumpers (TERM_CAN_x), the battery (upper big circle) and the battery –low LED (on top the CANbus connectors).

The pin-header near the upper chipset is for factory programming only.

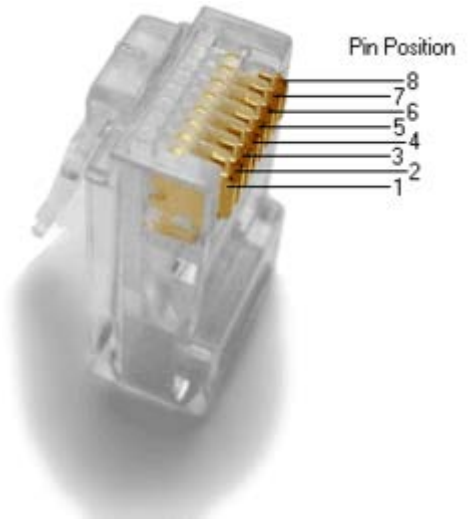
CANBUS CONNECTORS

The four CANbus connectors on the front bezel have a standard RJ45 format with the pin assignment as described here below:



RJ-45 connector

SOCKET PIN POSITION



PLUG PIN POSITION

RJ45 CAN Bus PinOut

RJ45 Pin #	Signal Name	Signal Description
1	CAN_H	Dominant High
2	CAN_L	Dominant Low
3	CAN_GND	Ground
4	Not used	
5	Not used	
6	CAN_SHLD	CAN Shield, Optional
7	CAN_GND	Ground
8	Not used	

TERMINATIONS

Each Jumper “TERM_CAN_x” allows for setting the termination resistor for that channel.
The Jumper setting is as follows:

1-2 = Terminated
2-3 = NOT Terminated

Set the termination only if the board is at one end of the CANbus connection. At opposite end, the termination will be set only on the furthest peripheral device.

BATTERY OPERATION

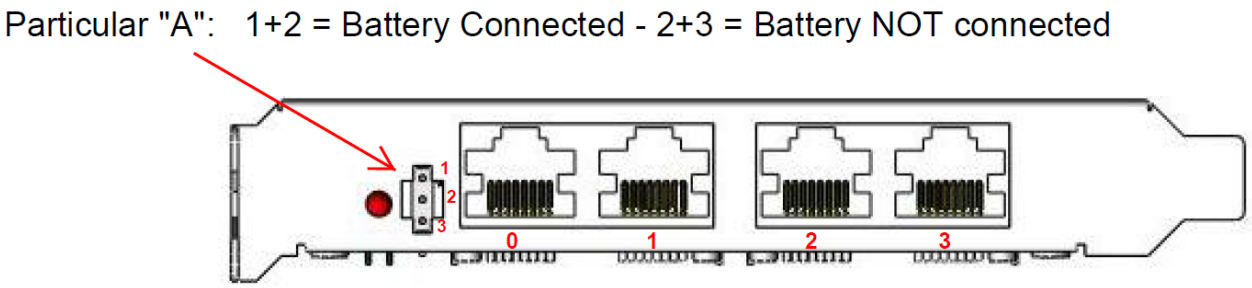
WARNING:

This product contains a Lithium battery. You should pay attention regarding the handling and disposal of the board. Part of the circuit is under power even when the PC is off or the board is removed.

Placing the board in contact with metallic surfaces or parts may cause permanent damage, explosion or fire.

The board is shipped with the battery isolated from the circuit since front Jumper is set 2+3. To connect battery (used to power on the static RAM only) requires the jumper is set 1+2.

Set this jumper only just before final installation and use.



Following pages describe the board's architecture from a resources point of view.

PROGRAMMING

The CRP board is a PCI based board and supports all PCI features up to version 2.1. The PCI BIOS, cooperating with Plug & Play operating system should assign resources to the board through standard mechanism. The driver should then read the assigned addresses and numbers back from the standard PCI configuration space.

Configuration space

The configuration space is the following:

PCI CFG Register Address	To ensure software compatibility with other versions of PCI 9050 family and to ensure compatibility with future enhancements, write "0" to all unused bits.				PCI Writable	Serial EEPROM Writable
	31 24	23 16	15 8	7 0		
00h	Device ID		Vendor ID		N	Y
04h	Status		Command		Y	N
08h	Class Code			Revision ID	N	Y[31:8]
0Ch	BIST	Header Type	PCI Latency Timer	Cache Line Size	Y[7:0]	N
10h	PCI Base Address 0 for Memory Mapped Configuration Registers				Y	N
14h	PCI Base Address 1 for I/O Mapped Configuration Registers				Y	N
18h	PCI Base Address 2 for Local Address Space 0				Y	N
1Ch	PCI Base Address 3 for Local Address Space 1				Y	N
20h	PCI Base Address 4 for Local Address Space 2				Y	N
24h	PCI Base Address 5 for Local Address Space 3				Y	N
28h	Cardbus CIS Pointer (Not Supported)				N	N
2Ch	Subsystem ID		Subsystem Vendor ID		N	Y
30h	PCI Base Address for Local Expansion ROM				Y	N
34h	Reserved				N	N
38h	Reserved				N	N
3Ch	Max_Lat	Min_Gnt	Interrupt Pin	Interrupt Line	Y[7:0]	Y[15:8]

All standard PCI operations do apply for reading and writing to PCI configuration space. The CRP board supports the complete configuration space as per PCISIG specifications. As different operating systems may map the configuration space differently, the programmer should follow the indication for the operating system in use to read and write into the configuration space, as well as map physical addresses into virtual addresses for use by the driver or applications.

The following are the relevant registers normally used by a device driver to identify the board and to configure it:

DETAIL OF REGISTERS

Offset (hex)	Size (bits)	Name	Description
02h	16	Device ID	0x9052
00h	16	Vendor ID	0x10b5
2Eh	16	Subsystem ID	0x3062
2Ch	16	Subsystem Vendor ID	0x10b5
10h	32	PCI Base Address 0	Memory Address of device Local Registers
14h	32	PCI Base Address 1	IO Address of device Local Registers (Not used)
18h	32	PCI Base Address 2	Memory Address of CAN controllers
1Ch	32	PCI Base Address 3	Memory Address of Battery backed RAM
20h	32	PCI Base Address 4	Unused
24h	32	PCI Base Address 5	Unused
3Ch	8	Interrupt Line	System interrupt assigned to board devices
04h	16	Command	PCI bus command word
06h	16	Status	PCI bus status word

If the operating system supports plug & play, at the time of driver loading the various addresses and the interrupt line are already assigned. It should be remembered that the assigned addresses are physical addresses; normally the drivers and applications live in virtual address space. It is therefore necessary to map the physical address into the virtual space of the driver or application. This is an operation normally supported by operating systems, but may be different for each, so consult the operating system manual. Also depending on operating systems, the Status and Command word may be accessed to enable the board mapping. Normally only memory access can be used, but optionally the Local Registers may be accessed through I/O also.
Once mapped into virtual addresses, the internal CRP resources can be accessed directly through these addresses.
Starting from the address which is specified in register 18h (BAR2) the 4 CAN controllers are mapped. Each Controller occupies 256 bytes, at 1kB distance. That is, each controller gets the first 256 bytes of its dedicated 1kB space.
The registers which have been evidenced above should be set by BIOS after a boot. Then the O.S. can make a BIOS call to have the actual PCI configuration.

PCI Address Space 0 and 1

PCI address Space 0 and 1 are dedicated to the Plx9052 internal registers called Local Registers.
The two spaces map the same registers. The only difference is that the space 0 must be accessed using memory access, and space 1 using I/O access. Some host architectures differentiate between the two access types, and normally the operating system provides suitable mappings and access routines. It is not necessary to use both spaces because they accomplish the same purpose. The choice may be the personal preference, or performance considerations (IO normally is slower that memory access).

PCI Address Space 2

PCI Address Space 2 is configured to map the four CAN controllers in the same space. To accommodate all resources, the space should be at least mapped into a 4KB memory address range.

The 4KB space used is divided internally into 4 sub-ranges of 1KB each where the following resources are mapped:

Offset	Size	Name	Description
0KB	256B	CAN 0	Can controller 0 internal registers
1KB	256B	CAN 1	Can controller 1 internal registers
2KB	256B	CAN 2	Can controller 2 internal registers
3KB	256B	CAN 3	Can controller 3 internal registers
4KB	1Byte	Digital Inputs	Digital Input Register (to be used in upgraded version of the board)
5KB	1Byte	Not used	Not used

Within the 6KB space there are unused ranges. These addresses are reserved and must not be used. Generating addresses in these ranges may affect the normal behavior of the other devices that share the same space. Actually some sort or duplication of device images through the unused space is possible.

PCI Address Space 3

PCI Address Space 3 maps the internal battery backed Static RAM. The size is always 1MB made by 2 x 512kB Low-Low power RAM chips.

Interrupts

The CRP board supports one standard PCI interrupt (# INTA).

Local Registers

The content of the local registers is used to configure the hardware resources internally to the PCI interface chip and to map the board resources into internal address map. These registers are automatically loaded at power on and generally the driver should not change them.

Only two 32 bit registers at offset 4Ch (INTCSR) and 50h (CNTRL) of local registers must be used by the driver to manipulate board interrupts and to read from the internal E2PROM.

In these registers there are bit fields with different meanings; the software must be careful not to change any unwanted field. Failure to do so may result in malfunctioning or even complete PC lock.

INTCSR (4Ch)

This register is used to control the CRP interrupts behavior.
The driver, in order to enable and check for pending interrupts, must manipulate some bits of this register; other bits should reflect the internal routing and connection and must be initialized with the suggested value before using any interrupts.

This is the layout of the Interrupt control register at offset 4Ch into the local registers:

Bit(s)	Description	Notes
0	Local Interrupt 1 Enable. Value of 1 indicates enabled. Value of 0 indicates disabled.	The driver should turn on this bit only after having installed an interrupt service routine.
1	Local Interrupt 1 Polarity. Value of 1 indicates Active high. Value of 0 indicates Active low.	This must be set to zero.
2	Local Interrupt 1 Status. Value of 1 indicates Interrupt active. Value of 0 indicates Interrupt not active.	The driver should check this bit to see if the controllers tied to this line need attention.
3		Not used
4		Not used
5		Not used
6	PCI Interrupt Enable. Value of 1 enables PCI interrupt.	This is a general interrupt enable bit. The driver should turn on this bit only after having installed an interrupt service routine.
7	Software Interrupt. Value of 1 generates interrupt.	This may be used for testing purposes
8	Local Interrupt 1 Select Enable. Value of 1 indicates enabled edge triggerable interrupt. Value of 0 indicates enabled level triggerable interrupt.	This must be set to zero.
9		Not used.
10	Local Edge Triggerable Interrupt Clear Bit. Writing 1 to this bit clears Interrupt_1.	Not used.
11		Not used.
12	ISA Interface Mode Enable. Writing 1 enables ISA Interface mode. Writing 0 disables ISA Interface mode.	This must be set to zero.
31:13	Unused.	

CNTRL (50h)

The CNTRL register contains several bits to configure the PCI interface chip. Normally these settings are automatically loaded during power on, and should not be changed by the driver.
An internal EEPROM contains the entire chip configuration, and it is programmed at factory. This internal EEPROM contains also the board serial number. If the driver wants to read this information for tracking and diagnostic purposes, it must use bits 24 to 28 of this register.

The driver should be careful not to write into other registers or in the EEPROM itself, or the PC may lock or even may not reboot anymore, and the CRP board must be repaired at factory.

Also changing other reserved bits may result in a PC lock. It is recommended that the register is read first and then the setting or resetting of individual bits is made by bitwise operations using appropriate masks, in order to leave the reserved bits untouched, and then written back.

Bit(s)	Description	Notes
0:23	Reserved	Do Not Change
24	Serial EEPROM Clock for Local or PCI Bus Reads or Writes to serial EEPROM. Toggling this bit generates a serial EEPROM clock.	EEP_CLK
25	Serial EEPROM Chip Select. For local or PCI bus reads or writes to serial EEPROM, setting this bit to 1 provides serial EEPROM chip select active.	EEP_CS
26	Write Bit to serial EEPROM. For writes, this output bit is the input to serial EEPROM. Clocked into the serial EEPROM by serial EEPROM clock.	EEP_D
27	Read serial EEPROM Data Bit. For reads, this input bit is the output of serial EEPROM. Clocked out of the serial EEPROM by serial EEPROM clock.	EEP_Q
28:31	Reserved	Do Not Change

The EEPROM is organized into 16bits words, and a complete word must be read each time.

To read one EEPROM word, the following sequence should be followed:

- ⇨ Assure that EEP_CS and EEP_CLK are zero and the EEP_D is one
- ⇨ Turn EEP_CS to one
- ⇨ Set EEP_D to one (start bit) and pulse the EEP_CLK high and then low
- ⇨ Send the read command (10b) by setting EEP_D to one, pulsing the EEP_CLK, then setting the EEP_D to zero and pulsing the EEP_CLK again.
- ⇨ Send the word address you want to read by setting the EEP_D line according to each address bit (MSB first) and pulsing the EEP_CLK line for each bit. The address is a 6bit value, so send A5 to A0 and pulse the clock six times.
- ⇨ Check if the EEPROM has understood the command and is ready to answer by checking the EEP_Q bit that should be zero. A small delay should be required to allow the EEPROM to respond. A maximum of 700nS is specified on the EEPROM data sheet.
- ⇨ Pulse the EEP_CLK line to shift out the first (MSB) bit of the requested word, and read EEP_D into bit 15 of the result word.
- ⇨ Repeat pulsing EEP_CLK and reading the next lesser significant bit for a total of sixteen times to build the complete result word.
- ⇨ Remove the EEP_CS setting it to zero, and optionally prepare the EEP_D bit for the next reading (to 1).

EEPROM contents

The EEPROM words relevant for the driver are the four words located at addresses 3Ch to 3Fh. Usually these words are read together into a memory array or structure and then accessed by the s/w using bit fields or byte offset. The contents of this memory structure are as follows, where the offset is intended from the beginning.

Offset	Bits	Description
0		Not used
0		Not used
1		Not used
1		Not used
2	7:0	Memory size : value = 06h for 1MB
3	7:0	Reserved
4	7:0	Card Serial number (low)
5	7:0	Card Serial number (high)
6	**7:0**	**Board type (default 20 decimal for CRP)**
7	**7:0**	**Low byte of the CRC 16 of bytes 0 to 6**

The following meaning applies to the speed and size settings:

Can Speed value	Description
0	CAN controller is not present on board.
1	20Kbit/S
2	62.5Kbit/S
3	125Kbit/S
4	250Kbit/S
5	500Kbit/S
6	800Kbit/S
7	1Mbit/S

Memory size value	Description
-	-
-	-
6	1Mbyte

Can controller

All CAN controllers share the same address space (see above), and use a sub-space of 256 bytes where different internal registers are mapped. For additional details on how to configure the CAN controller chip, please refer to the SJA1000 Data Sheet.

The four Can controllers are independent and can operate separately, but two elements tie the controllers together, and the programmer must be aware of the consequences:

- ⇨ Interrupt line. All 4 controllers share a common interrupt line, which is activated by both Can controllers in a wired-OR fashion. The software should check each controller internal registers to determine the cause of interrupt and correctly manage the case in which no interrupt are really found on a controller, because the cause of interrupt may be in the other one.

- ⇨ Master Clock Input. All four controllers share a single 16Mhz quartz (not oscillator!), which is connected to the first Can controller, while the other Can controllers clock input is connected to the clock out pin of the previous one. This means that the setup program must enable the clock output of the preceding controllers in order to have the following controller working, even if the preceding controllers are not used.

This means that we can use a cheap quartz (not oscillator) and make a daisy chain for the clock signal, from the 1st device to the following device and so on for all the controllers.

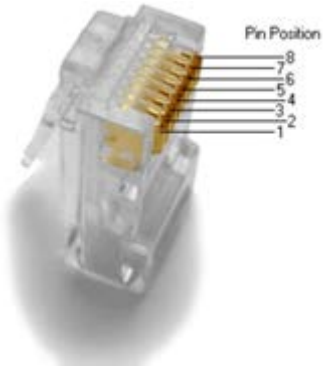
Memory

The 1 M B Static RAM memory has its independent memory space.
The memory is mapped in the physical space described by the PCI base register 3 corresponding to the internal local address space 1. This space must be mapped into a virtual space of the same size. The memory size is specified in the EEPROM. (value = 6 for 1MB).
The memory have a physical layout of 16bits, but must be accessible both for reading and writing in Bytes, Words, or Long Words chunks.

CABLE ADAPTOR TO CHANGE RJ45 TO DSUB9M

An adaptor cable can be purchased from CJB, to change the RJ45 into DSUB9 Male connectors.

CABLE LENGTH = 10 cm

SHIELDED RJ45 PLUG pin #	Description		DSUB9M pin	Description
				
1	CAN-H	⇔	7	CAN-H
2	CAN-L	⇔	2	CAN-L
3 & 7	CAN-GND	⇔	3	CAN-GND
6	CAN-SHLD		5	CAN-SHLD
RJ45 SHIELD	Shield (body of RJ45 Connector)	⇔	DSUB9M SHIELD	Shield (body of DSUB Connector)

© CJB 2012